



Development of Non-sequential Ray-tracing Software for Cosmic-ray Telescopes

AKIRA OKUMURA^{1,2}, MASAOKI HAYASHIDA^{2,3}, HIDEAKI KATAGIRI⁴, TAKAYUKI SAITO⁵, VLADIMIR VASSILIEV⁶

¹*Institute of Space and Astronautical Science, JAXA, Sagami-hara, Kanagawa 252-5210, Japan*

²*Kavli Institute for Particle Astrophysics and Cosmology, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA*

³*Department of Astronomy, Graduate School of Science, Kyoto University, Sakyo-ku, Kyoto 606-8502, Japan*

⁴*College of Science, Ibaraki University, Mito 310-8512, Japan*

⁵*Max-Planck-Institut für Physik, Föhringer Ring 6, 80809 München, Germany*

⁶*Department of Physics and Astronomy, University of California Los Angeles, CA 90095-1547, USA*

oxon@astro.isas.jaxa.jp (Akira Okumura)

Abstract: We have developed non-sequential ray-tracing software which is aimed to be widely used, along with air-shower simulations, in the design of optical systems for cosmic-ray experiments. The code is based on the ROOT geometry library to provide a non-sequential photon tracking system, which is valuable when simulating refraction and multiple reflections. In addition to the basic ROOT classes, we have implemented new geometry ROOT classes so that users can flexibly define various geometries such as aspherical or Winston-cone type surfaces. We demonstrate the capabilities and performance of the software with examples of optical systems used in current and future experiments.

Keywords: Ray tracing, Software, Optical System, Mirror, Lens, Light Guide, ROOT

1 Introduction

Optical systems for modern cosmic-ray (CR) experiments play an important role to improve their detection sensitivities of very-high-energy (VHE) gamma rays or ultra-high-energy (UHE) CRs. The basic strategy in the design of an optical system is to simultaneously achieve a large effective area of mirrors ($\sim 5 \text{ m}^2$ to $\sim 500 \text{ m}^2$), a wide field of view ($\sim 5^\circ$ to $\sim 50^\circ$ in diameter), a high angular resolution ($\sim 0.1^\circ$ to $\sim 1^\circ$), and low cost per telescope. In order to satisfy these requirements that are not demanded for optical-to-infrared telescopes, some techniques such as segmented mirrors and light guides are widely used for CR telescopes. Consequently optical simulations of CR telescopes become more puzzling, as their optical systems get more complicated.

Ray-tracing software is often developed independently in each CR experiment or institute. This is because existing programs for optical telescopes are not adequate for CR telescopes (see details in Subsection 2.1). However, many requirements in optical simulations are common in all CR experiments. This means that development of a convenient and accurate ray-tracing program shared by the CR community will enable us to save the time for software development and debugging. Our objective in the present paper is to offer a standard ray-tracing program to be used quite

easily for many optical studies in CR experiments, and to reduce dispensable effort in parallel software development.

2 ROBAST

We have developed a new ray-tracing program, ROBAST¹ (ROot-BASed Simulator for ray Tracing), which is based on the ROOT framework [1]. ROBAST fully utilizes its geometry library [2] to offer several functionalities required for ray-tracing simulations of CR telescopes.

2.1 Requirements and Functionalities

2.1.1 Non-sequential Ray Tracing

The normal ray-tracing method adopted in most programs is so-called “sequential ray tracing” in which the order of all surfaces in an optical system is defined by the user, and photons reach the surfaces sequentially. It is often used for simulations of simple optical systems, such as a single-dish parabolic mirror. On the contrary, we need “non-sequential ray-tracing” mode to simulate scattered photons or multiple segmented mirrors. This is because the order of multiple reflections and refraction cannot be specified uniquely by the user.

1. <http://sourceforge.net/projects/robast/>

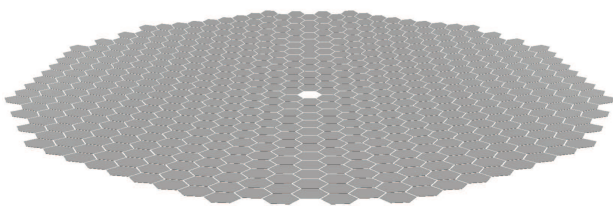


Figure 1: An example of hexagonal segmented mirrors aligned on a parabolic surface. The same optical system will be used for the large size telescopes in the Cherenkov Telescope Array project [11].

Segmented mirrors are almost always used in CR experiments to enlarge the effective mirror area of each telescope, and to reduce the total cost per area. Each mirror segment is usually placed on a parabolic or spherical surface, or aligned on the Davies-Cotton configuration as shown in Figure 1 [3, 4, 5, 6, 7, 8]. These mirrors cannot be simulated by sequential ray-tracing software, because the intersection point of an input photon with each mirror segment must be calculated. The non-sequential mode is also required when we simulate light guides such as Winston cones [9] or similar surfaces, because photons can be reflected by light guides multiple times [10, 6].

Coding a non-sequential ray-tracing algorithm from scratch is not simple, but it has been already established as particle tracking systems in high-energy physics. Since powerful particle tracking systems are already available in ROOT [2] and Geant4 [12], we have decided to use the ROOT geometry library as the ray tracer in ROBAST. The library also enables us to satisfy other requirements.

2.1.2 Flexibility of Geometry Construction

As shown in Figure 1, hexagonal or rectangle segmented mirrors are often used [7, 8] as well as circular mirrors [10, 6]. Each segment shape can be easily built using basic geometry shapes² and “boolean” operations offered by ROOT. For example, a hexagonal mirror can be defined by an “AND” operation of a sphere and a hexagonal cylinder as illustrated in Figure 2. In addition to mirror shapes, other optical components and telescope structures can be built. This functionality is important for accurate calculations of effective areas and shadowing effects of telescopes.

The shape of a hexagonal Winston cone or a similar light guide is mathematically expressed by a combination of simple geometries, such as paraboloids and cones. However, due to the limited numbers of ROOT geometry shapes, even a basic hexagonal Winston cone cannot be built with the default ROOT classes. In addition, some optical systems recently designed use more complicated shapes that are not available in ROOT [13, 14]. We have added two more geometry classes, `AGeoWinstonConePoly` and `AGeoAsphericDisk`, in ROBAST so that users can build complex geometries as demonstrated in Figures 3 and

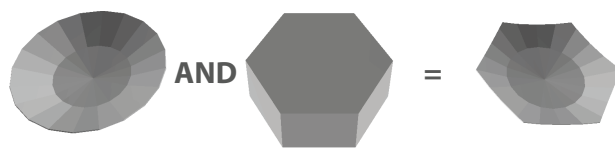


Figure 2: An example of a mirror segment built with a “boolean” operation of a partial sphere and a hexagonal cylinder. The region spatially shared with two objects is used as a new object.

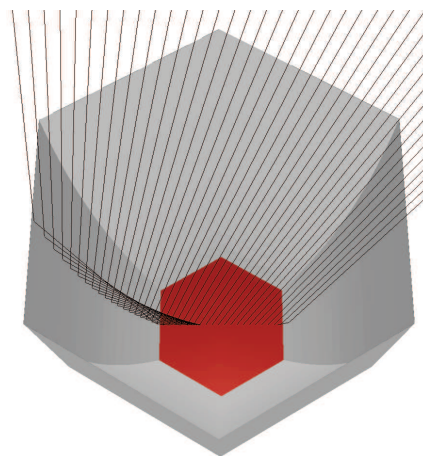


Figure 3: An example of a hexagonal Winston cone built with `AGeoWinstonConePoly` class consisting of six inclined paraboloids. The input and output radii are 20 mm and 10 mm, respectively. Photon tracks simulated by `ARay` class instances are shown with the black lines. The red hexagon imitates the photocathode of a photomultiplier tube (PMT).

4. Of course new geometry classes can be easily implemented by the user to build more complicated shapes.

It is also easy for a user to place an optical component at any coordinates with any angles using translation and rotation matrices. This feature is important when the user evaluates the degradation of optical performance caused by misalignment of optical components.

2.1.3 Visualization

As already shown in Figures 1 to 4, ROBAST is able to visualize geometries thanks to the ROOT OpenGL viewer. Visualization plays a major role when user-defined geometries are inspected by eye. However, visualization functionality has not been well integrated for self-made programs because it is difficult to develop a new visualization system from scratch. The built-in visualization engine will also be helpful for presentations.

2. See e.g. <http://root.cern.ch/root/html/doc/TGeoBBox.htm> and derived classes.

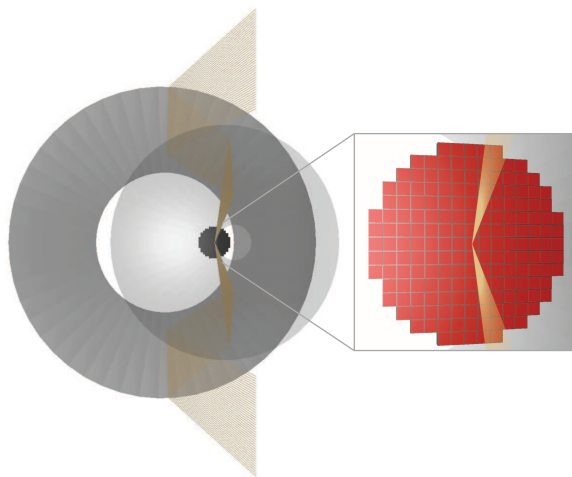


Figure 4: The Schwarzschild-Couder optical system [14] built with `AGeoAsphericDisk` class. The primary (larger doughnut) and the secondary (semitransparent smaller doughnut) mirrors are approximated by 14th-order polynomials. 177 multi-anode PMTs (red squares) are spherically aligned on the focal surface. The diameters of the primary mirror, the secondary mirror, and the focal plane are, 9.7 m, 5.4 m, and 0.78 m, respectively.

2.1.4 Connectivity with Other Software

ROBAST has been developed especially for CR telescope simulations. It depends on other external software. For example, ray-tracing software is frequently used with air-shower Monte Carlo simulations or electronics simulations in numerous batch jobs. In addition, ROBAST results are expected to be analyzed seamlessly by standard tools such as ROOT (CINT or C++) or Python libraries. Since ROBAST runs in either compiled C++ code, ROOT CINT scripts, or Python scripts, most CR researchers can immediately use it with other external software. On the other hand, existing ray-tracing programs will not generally work this way.

2.1.5 Multi Platform Support

ROBAST runs on all of the three major operation systems (OSs) used by high-energy astrophysicists: Mac OS X, Linux, and Windows.

2.2 Comparison with Other Programs

In the previous subsection, we listed requirements for ray-tracing software and described the advantages that come with ROBAST. In this subsection, we compare ROBAST with other major programs, and describe some strengths and weaknesses.

2.2.1 Geant4

Geant4 is a simulation toolkit for particle tracking and interaction [12]. It is able to track optical photons as well as X-rays and gamma rays so that a user can simulate scintillation detectors. The optical photon tracking in Geant4 is similar to that of ROBAST, while it has two main advantages which are not currently available in ROBAST.

The first advantage is that Geant4 is able to simulate the roughness of optical boundaries, and to choose the reflection angle distribution from a few options. Using these capabilities, a user can build more realistic optical components, e.g. a crystal scintillator wrapped in a diffusive reflector. Since this functionality is critical to simulate the mirror roughness, it will be implemented in the next release of ROBAST.

The other is that Geant4 can import very complicated volumes defined in a GDML [15] file along with `G4TessellatedSolid` class which consists of multiple polygon facets. ROOT and ROBAST are also able to import GDML files, but they do not have an equivalent of `G4TessellatedSolid`. Therefore, Geant4 is preferable if a user needs to convert his/her complicated CAD file into GDML.

If a user is more familiar with Geant4 than ROOT, the user can select it for ray-tracing simulation. But it does not have similar classes as `AGeoAsphericDisk` and `AGeoWinstonConePoly` which are quite important for CR telescopes. In addition, ROOT is more widely used than Geant4.

2.2.2 ZEMAX

There are many commercial or open source ray-tracing programs which are not being developed especially for CR telescopes. A commercial software, ZEMAX³ which runs on Windows only, is one of the most famous programs used in many aspects of optical simulations. The advantages of the sophisticated commercial software are as follows.

First, it has a built-in optimization engine. The surface of optical components can be very quickly optimized from its graphical interface, especially in the non-sequential mode. But ROBAST does not have any optimization engine specifically designed for optical systems.

Second, ZEMAX has many built-in analysis tools. However, optical systems used in CR experiments are less precise than other systems. For example, our optical systems do not reach the diffraction limit, and thus, we do not need high performance tools in most situations.

ZEMAX or similar software would be better than ROBAST at the first stage of designing optical systems. However, ROBAST will suit all our requirements after optimizations. An advanced ROBAST user can build own optimizer using MINUIT distributed with ROOT.

3. <http://www.zemax.com/>

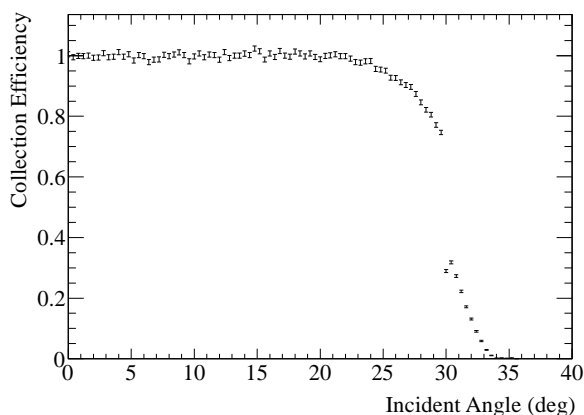


Figure 5: The collection efficiency of the hexagonal Winston cone shown in Figure 3. $\sim 14,000$ random photons were simulated for each incident angle. The mirror reflectivity was assumed to be 100%. The discontinuity at $\sim 30^\circ$ is due to the two parabolic surfaces vertical to the incident photons in the figure.

3 Simulation Examples

Figure 5 is a simulation example demonstrating the collection efficiency of the Winston cone shown in Figure 3. In producing this simulation, the photon tracking is transparent to the user who only has to build the cone's shape in a ROOT script.

Figure 6 shows the spot diagrams of the Schwarzschild-Couder optical system shown in Figure 4. The photon coordinates and directions on the focal plane are available within the same ROOT script that produced this simulation example. It is easy to produce these spot diagrams and the incident angle distribution on the focal plane on the fly with the ROOT analysis classes.

4 Conclusion

We have developed a non-sequential ray-tracing program which can be used in many aspects of CR experiments. ROBAST satisfies many requirements for studies of CR telescopes. It is already being used for some optical studies for the CTA and the MAGIC experiments. We expect ROBAST to be also used in other experiments in near future.

5 Acknowledgements

We are grateful to Dr. Andrei Gheata who is one of the maintainers of the ROOT geometry library. He kindly answered many questions at the first stage of the ROBAST development. We could not start this project without the ROOT geometry library and his great support. We would also like to thank Dr. Akito Kusaka who gave A. O. an ini-

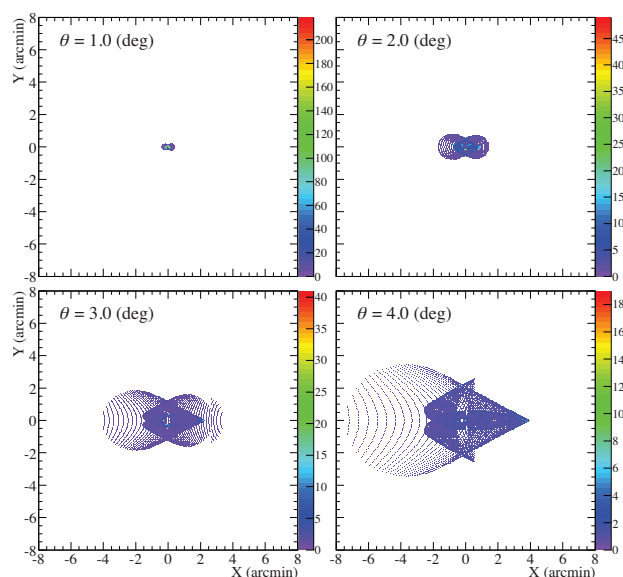


Figure 6: Spot diagrams of the Schwarzschild-Couder optical system shown in Figure 4.

tial idea of ray-tracing code. A. O. and M. H. are supported by Grant-in-Aid for JSPS fellows.

References

- [1] R. Brun and F. Rademakers, Nucl. Instr. Meth. Phys. Res. A **389**, 81 (1997).
- [2] R. Brun, A. Gheata, and M. Gheata, Nucl. Instr. Meth. Phys. Res. A **502**, 676 (2003).
- [3] J. M. Davies and E. S. Cotton, The Journal of Solar Energy Science and Engineering **1**, 16 (1957).
- [4] T. C. Weekes *et al.*, Astrophys. Jour. **342**, 379 (1989).
- [5] A. Kawachi *et al.*, Astropart. Phys. **14**, 261 (2001).
- [6] K. Bernlöhner *et al.*, Astropart. Phys. **20**, 111 (2003).
- [7] J. Holder *et al.*, Astropart. Phys. **25**, 391 (2006).
- [8] M. A. L. de Oliveira *et al.*, Nucl. Instr. Meth. Phys. Res. A **522**, 360 (2004).
- [9] R. Winston, Journal of the Optical Society of America **60**, 245 (1970).
- [10] S. Kabuki *et al.*, Nucl. Instr. Meth. Phys. Res. A **500**, 318 (2003).
- [11] The CTA Consortium, arXiv 1008.3703, (2010).
- [12] S. Agostinelli *et al.*, Nucl. Instr. Meth. Phys. Res. A **506**, 250 (2003).
- [13] M. Sasaki, A. Kusaka, and Y. Asaoka, Nucl. Instr. Meth. Phys. Res. A **492**, 49 (2002).
- [14] V. Vassiliev, S. Fegan, and P. Brousseau, Astropart. Phys. **28**, 10 (2007).
- [15] R. Chytrcek *et al.*, Nuclear Science, IEEE Transactions on **53**, 2892 (2006).